

(51) Int.Cl. ⁷	識別記号	F I	テマート* (参考)
H 0 4 L 9/32		H 0 4 L 9/00	6 7 3 A
G 0 6 F 15/00	3 3 0	G 0 6 F 15/00	3 3 0 E
H 0 4 L 9/08		H 0 4 L 9/00	6 0 1 C
			6 7 5 B

審査請求 未請求 請求項の数66 O L (全 16 頁)

(21) 出願番号	特願2000-210117(P2000-210117)	(71) 出願人	596077259 ルーセント テクノロジーズ インコーポ レイテッド Lucent Technologies Inc. アメリカ合衆国 07974 ニュージャージー ー、マレーヒル、マウンテン アベニュー 600-700
(22) 出願日	平成12年7月11日(2000.7.11)	(74) 代理人	100081053 弁理士 三俣 弘文
(31) 優先権主張番号	0 9 / 3 5 3 4 6 8		
(32) 優先日	平成11年7月13日(1999.7.13)		
(33) 優先権主張国	米国 (U S)		

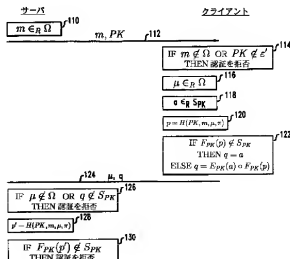
最終頁に続く

(54) 【発明の名称】 相互ネットワーク認証方法

(57) 【要約】

【課題】 安全性を証明可能なパスワード単独相互認証プロトコルを実現する。

【解決手段】 クライアントは、サーバから受信した公開鍵が、公開鍵暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定する。そのような要素でない場合、認証はクライアントによって拒否され、そうでない場合、プロトコルは続行される。一実施例では、クライアントとサーバはいずれも、認証目的で使用される1つのパスワードを所有する。クライアントは、少なくとも公開鍵およびパスワードの関数としてパラメータ p を生成する。公開鍵空間マッピング関数 F_{PK} を p に作用させた結果 $F_{PK}(p)$ が公開鍵のメッセージ空間の要素である場合、クライアントが、公開鍵を用いて公開鍵のメッセージ空間の実質的にランダムな要素を暗号化し、その結果と、 $F_{PK}(p)$ との間で、公開鍵メッセージ空間の群演算を実行する。



【特許請求の範囲】

【請求項 1】 公開鍵暗号方式を利用したクライアントとサーバの間の相互ネットワーク認証方法において、前記クライアントが、

前記サーバから公開鍵を受信するステップと、

前記公開鍵が、前記公開鍵暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップと、

前記公開鍵が前記テスト可能スーパーセットの要素でない場合、認証を拒否するステップとを有することを特徴とする相互ネットワーク認証方法。

【請求項 2】 前記クライアントが、

前記公開鍵が前記テスト可能スーパーセットの要素である場合、

少なくとも前記公開鍵およびパスワードの関数としてパラメータ p を生成するステップと、

公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

$F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素でない場合、

パラメータ q を、前記公開鍵のメッセージ空間の実質的にランダムな要素とするステップと、

q を前記サーバに送信するステップとをさらに有することを特徴とする請求項 1 に記載の方法。

【請求項 3】 p は、さらに、少なくとも、前記サーバから受信したパラメータ m の関数として生成されることを特徴とする請求項 2 に記載の方法。

【請求項 4】 p は、さらに、少なくとも、実質的にランダムな数 μ の関数として生成されることを特徴とする請求項 3 に記載の方法。

【請求項 5】 前記クライアントが、

前記公開鍵が前記テスト可能スーパーセットの要素である場合、

少なくとも前記公開鍵およびパスワードの関数としてパラメータ p を生成するステップと、

公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

$F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素である場合、

前記公開鍵を用いて前記公開鍵のメッセージ空間の実質的にランダムな要素を暗号化し、その結果と $F_{pk}(p)$ との間で、公開鍵メッセージ空間の群演算を実行することによって、パラメータ q を生成するステップと、

q を前記サーバに送信するステップとをさらに有することを特徴とする請求項 1 に記載の方法。

【請求項 6】 p は、さらに、少なくとも、前記サーバから受信したパラメータ m の関数として生成されることを特徴とする請求項 5 に記載の方法。

【請求項 7】 p は、さらに、少なくとも、実質的にランダムな数 μ の関数として生成されることを特徴とする請求項 6 に記載の方法。

【請求項 8】 m および μ をパラメータとしてディフィ・ヘルマンプロトコルを用いてセッション鍵を生成するステップをさらに有することを特徴とする請求項 7 に記載の方法。

【請求項 9】 前記クライアントが、

前記公開鍵が前記テスト可能スーパーセットの要素である場合、

少なくとも前記公開鍵とパスワードの関数との関数としてパラメータ p を生成するステップと、

公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

$F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素でない場合、

パラメータ q を、前記公開鍵のメッセージ空間の実質的にランダムな要素とするステップと、

q を前記サーバに送信するステップとをさらに有することを特徴とする請求項 1 に記載の方法。

【請求項 10】 前記パスワードの関数は、前記パスワードの方向性ハッシュ関数であることを特徴とする請求項 9 に記載の方法。

【請求項 11】 p は、さらに、少なくとも、前記サーバから受信したパラメータ m の関数として生成されることを特徴とする請求項 9 に記載の方法。

【請求項 12】 p は、さらに、少なくとも、実質的にランダムな数 μ の関数として生成されることを特徴とする請求項 11 に記載の方法。

【請求項 13】 前記クライアントが、

前記公開鍵が前記テスト可能スーパーセットの要素である場合、

少なくとも前記公開鍵とパスワードの関数との関数としてパラメータ p を生成するステップと、

公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

$F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素である場合、

前記公開鍵を用いて前記公開鍵のメッセージ空間の実質的にランダムな要素を暗号化し、その結果と $F_{pk}(p)$ との間で、公開鍵メッセージ空間の群演算を実行することによって、パラメータ q を生成するステップと、

q を前記サーバに送信するステップとをさらに有することを特徴とする請求項 1 に記載の方法。

【請求項 14】 前記パスワードの関数は、前記パスワードの方向性ハッシュ関数であることを特徴とする請求項 13 に記載の方法。

【請求項 15】 p は、さらに、少なくとも、前記サーバから受信したパラメータ m の関数として生成されることを特徴とする請求項 13 に記載の方法。

【請求項 16】 p は、さらに、少なくとも、実質的にランダムな数 μ の関数として生成されることを特徴とする請求項 15 に記載の方法。

【請求項 17】 m および μ をパラメータとしてディフィ・ヘルマンプロトコルを用いてセッション鍵を生成するステップをさらに有することを特徴とする請求項 16 に記載の方法。

【請求項 18】 公開鍵暗号方式を利用したクライアントとサーバの間の相互ネットワーク認証方法において、前記サーバが、

使用可能公開鍵暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である公開鍵をクライアントに送信するステップと、

前記クライアントから、パラメータ q として、前記公開鍵のメッセージ空間の要素を受信するステップとを有することを特徴とする相互ネットワーク認証方法。

【請求項 19】 前記クライアントで少なくとも前記公開鍵およびパスワードの関数として生成されたパラメータ p に、公開鍵空間マッピング関数 F_R を作用させた結果 $F_R(p)$ が、前記公開鍵のメッセージ空間の要素でない場合、

前記公開鍵のメッセージ空間の実質的にランダムな要素をパラメータ q として受信するステップを有することを特徴とする請求項 18 に記載の方法。

【請求項 20】 前記クライアントで少なくとも前記公開鍵およびパスワードの関数として生成されたパラメータ p に、公開鍵空間マッピング関数 F_R を作用させた結果 $F_R(p)$ が、前記公開鍵のメッセージ空間の要素である場合、

公開鍵メッセージ空間の実質的にランダムな要素を公開鍵暗号化した結果と、 $F_R(p)$ との間で、公開鍵メッセージ空間の群演算を実行した結果をパラメータ q として受信するステップを有することを特徴とする請求項 18 に記載の方法。

【請求項 21】 前記クライアントで少なくとも前記公開鍵とパスワードの関数との関数として生成されたパラメータ p に、公開鍵空間マッピング関数 F_R を作用させた結果 $F_R(p)$ が、前記公開鍵のメッセージ空間の要素でない場合、前記公開鍵のメッセージ空間の実質的にランダムな要素をパラメータ q として受信するステップを有することを特徴とする請求項 18 に記載の方法。

【請求項 22】 前記パスワードの関数は、前記パスワードの一方方向性ハッシュ関数であることを特徴とする請求項 21 に記載の方法。

【請求項 23】 前記クライアントで少なくとも前記公開鍵とパスワードの関数との関数として生成されたパラメータ p に、公開鍵空間マッピング関数 F_R を作用させ

た結果 $F_R(p)$ が、前記公開鍵のメッセージ空間の要素である場合、

公開鍵メッセージ空間の実質的にランダムな要素を公開鍵暗号化した結果と、 $F_R(p)$ との間で、公開鍵メッセージ空間の群演算を実行した結果をパラメータ q として受信するステップを有することを特徴とする請求項 18 に記載の方法。

【請求項 24】 前記パスワードの関数は、前記パスワードの一方方向性ハッシュ関数であることを特徴とする請求項 23 に記載の方法。

【請求項 25】 前記公開鍵暗号方式は RSA であり、前記公開鍵はパラメータ N および e からなり、前記公開鍵は、前記サーバによって、

N はある値より大きく、
 e は N より大きく、
 e は素数であるように選択されることを特徴とする請求項 18 に記載の方法。

【請求項 26】 前記公開鍵暗号方式は RSA であり、前記公開鍵はパラメータ N および e からなり、前記公開鍵は、前記サーバによって、

N はある値範囲内にあり、
 e はある値範囲内にあり、
 e は素数であるように選択されることを特徴とする請求項 18 に記載の方法。

【請求項 27】 前記公開鍵暗号方式は RSA であり、前記公開鍵はパラメータ N および e からなり、前記公開鍵は、前記サーバによって、

e は所定の値であり、
 N はある値範囲内にあるように選択されることを特徴とする請求項 18 に記載の方法。

【請求項 28】 RSA 暗号方式を利用したクライアントとサーバの間の相互ネットワーク認証方法において、前記クライアントが、

前記サーバから RSA 公開鍵 (N , e) を受信するステップと、

前記 RSA 公開鍵 (N , e) が、前記 RSA 暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップと、

前記 RSA 公開鍵 (N , e) が前記テスト可能スーパーセットの要素でない場合、認証を拒否するステップとを有することを特徴とする相互ネットワーク認証方法。

【請求項 29】 前記 RSA 公開鍵 (N , e) が、前記 RSA 暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップは、前記クライアントが、

N はある値より大きく、
 e は N より大きく、
 e は素数であるかどうかを決定するステップを含むことを特徴とする請求項 28 に記載の方法。

【請求項 30】 前記 RSA 公開鍵 (N , e) が、前記

RSA暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップは、前記クライアントが、
 N はある値範囲内にあり、
 e はある値範囲内にあり、
 e は素数であるかどうかを決定するステップを含むことを特徴とする請求項28に記載の方法。

【請求項31】 前記RSA公開鍵 (N , e) が、前記RSA暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップは、前記クライアントが、
 e は所定の値であり、
 N はある値範囲内にあるかどうかを決定するステップを含むことを特徴とする請求項28に記載の方法。

【請求項32】 前記クライアントが、前記RSA公開鍵 (N , e) が前記RSA暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である場合、
 少なくとも前記RSA公開鍵 (N , e) およびパスワードの関数としてパラメータ p を生成するステップと、
 公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
 $F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素でない場合、
 パラメータ q を、前記RSA公開鍵のメッセージ空間の実質的にランダムな要素とするステップと、
 q を前記サーバに送信するステップとをさらに有することを特徴とする請求項28に記載の方法。

【請求項33】 前記 $F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素であるかどうかを決定するステップは、
 p と N の最大公約数が1に等しいかどうかを決定するステップを含むことを特徴とする請求項32に記載の方法。

【請求項34】 p は、さらに、少なくとも、前記サーバから受信したパラメータ m の関数として生成されることを特徴とする請求項32に記載の方法。

【請求項35】 p は、さらに、少なくとも、実質的にランダムな数 μ の関数として生成されることを特徴とする請求項34に記載の方法。

【請求項36】 前記クライアントが、前記RSA公開鍵 (N , e) が前記RSA暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である場合、
 少なくとも前記RSA公開鍵 (N , e) およびパスワードの関数としてパラメータ p を生成するステップと、
 公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

$F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素である場合、
 a を、前記RSA公開鍵のメッセージ空間の実質的にランダムな要素として、 $q = (p \cdot a^e) \bmod N$ によりパラメータ q を生成するステップと、
 q を前記サーバに送信するステップとをさらに有することを特徴とする請求項28に記載の方法。

【請求項37】 前記 $F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素であるかどうかを決定するステップは、
 p と N の最大公約数が1に等しいかどうかを決定するステップを含むことを特徴とする請求項36に記載の方法。

【請求項38】 p は、さらに、少なくとも、前記サーバから受信したパラメータ m の関数として生成されることを特徴とする請求項36に記載の方法。

【請求項39】 p は、さらに、少なくとも、実質的にランダムな数 μ の関数として生成されることを特徴とする請求項38に記載の方法。

【請求項40】 m および μ をパラメータとしてディフィ・ヘルマンプロトコルを用いてセッション鍵を生成するステップをさらに有することを特徴とする請求項39に記載の方法。

【請求項41】 前記クライアントが、前記RSA公開鍵 (N , e) が前記RSA暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である場合、
 少なくとも前記RSA公開鍵 (N , e) とパスワードの関数との関数としてパラメータ p を生成するステップと、
 公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
 $F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素でない場合、
 パラメータ q を、前記RSA公開鍵のメッセージ空間の実質的にランダムな要素とするステップと、
 q を前記サーバに送信するステップとをさらに有することを特徴とする請求項28に記載の方法。

【請求項42】 前記パスワードの関数は、前記パスワードの方向性ハッシュ関数であることを特徴とする請求項41に記載の方法。

【請求項43】 前記 $F_{pk}(p)$ が前記RSA公開鍵のメッセージ空間の要素であるかどうかを決定するステップは、
 p と N の最大公約数が1に等しいかどうかを決定するステップを含むことを特徴とする請求項41に記載の方法。

【請求項44】 p は、さらに、少なくとも、前記サーバから受信したパラメータ m の関数として生成されるこ

とを特徴とする請求項 4 1 に記載の方法。

【請求項 4 5】 p は、さらに、少なくとも、実質的にランダムな数 μ の関数として生成されることを特徴とする請求項 4 4 に記載の方法。

【請求項 4 6】 前記クライアントが、前記 RSA 公開鍵 (N, e) が前記 RSA 暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素である場合、

少なくとも前記 RSA 公開鍵 (N, e) とパスワードの関数との関数としてパラメータ p を生成するステップと、

公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記 RSA 公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

$F_{pk}(p)$ が前記 RSA 公開鍵のメッセージ空間の要素である場合、

a を、前記 RSA 公開鍵のメッセージ空間の実質的にランダムな要素として、 $q = (p \cdot a^e) \bmod N$ によりパラメータ q を生成するステップと、

q を前記サーバに送信するステップとをさらに有することを特徴とする請求項 2 8 に記載の方法。

【請求項 4 7】 前記パスワードの関数は、前記パスワードの一方方向ハッシュ関数であることを特徴とする請求項 4 6 に記載の方法。

【請求項 4 8】 前記 $F_{pk}(p)$ が前記 RSA 公開鍵のメッセージ空間の要素であるかどうかを決定するステップは、

p と N の最大公約数が 1 に等しいかどうかを決定するステップを含むことを特徴とする請求項 4 6 に記載の方法。

【請求項 4 9】 p は、さらに、少なくとも、前記サーバから受信したパラメータ m の関数として生成されることを特徴とする請求項 4 6 に記載の方法。

【請求項 5 0】 p は、さらに、少なくとも、実質的にランダムな数 μ の関数として生成されることを特徴とする請求項 4 9 に記載の方法。

【請求項 5 1】 m および μ をパラメータとしてディフィ・ヘルマンプロトコルを用いてセッション鍵を生成するステップをさらに有することを特徴とする請求項 5 0 に記載の方法。

【請求項 5 2】 公開鍵暗号方式を利用したクライアントとサーバの間の相互認証方法において、

前記サーバが、公開鍵を前記クライアントに送信するステップと、

前記クライアントが、前記公開鍵が前記公開鍵暗号方式のすべての公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するステップと、

前記クライアントが、前記公開鍵が前記テスト可能スーパーセットの要素でないとき決定した場合、前記クライアントが、認証を拒否するステップとを有することを特徴

とする相互認証方法。

【請求項 5 3】 前記クライアントが、前記公開鍵が前記テスト可能スーパーセットの要素であると決定した場合、

前記クライアントが、少なくとも前記公開鍵およびパスワードの関数としてパラメータ p を生成するステップと、

前記クライアントが、公開鍵空間マッピング関数 F_{pk} を p に作用させた結果 $F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップとをさらに有することを特徴とする請求項 5 2 に記載の方法。

【請求項 5 4】 前記クライアントが、 $F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素であると決定した場合、

前記クライアントが、前記公開鍵を用いて前記公開鍵のメッセージ空間の実質的にランダムな要素 a を暗号化し、その結果と $F_{pk}(p)$ との間で、公開鍵メッセージ空間の群演算を実行することによって、パラメータ q を生成するステップと、

q を前記サーバに送信するステップとをさらに有することを特徴とする請求項 5 3 に記載の方法。

【請求項 5 5】 前記クライアントが、 $F_{pk}(p)$ が前記公開鍵のメッセージ空間の要素でないとき決定した場合、

前記クライアントが、パラメータ q を、前記公開鍵のメッセージ空間の実質的にランダムな要素とするステップをさらに有することを特徴とする請求項 5 4 に記載の方法。

【請求項 5 6】 前記サーバが、 q が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

前記サーバが、 q が前記公開鍵のメッセージ空間の要素でないとき決定した場合、前記サーバが、認証を拒否するステップと、

前記サーバが、 q が前記公開鍵のメッセージ空間の要素であると決定した場合、前記サーバが、前記公開鍵および前記パスワードの関数としてパラメータ p' を生成するステップと、

前記サーバが、公開鍵空間マッピング関数 F_{pk} を p' に作用させた結果 $F_{pk}(p')$ が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

前記サーバが、 $F_{pk}(p')$ が前記公開鍵のメッセージ空間の要素でないとき決定した場合、前記サーバが、認証を拒否するステップと、

前記サーバが、 $F_{pk}(p')$ が前記公開鍵のメッセージ空間の要素であると決定した場合、前記サーバが、 q と $F_{pk}(p')$ との間で、公開鍵メッセージ空間の群演算の逆を実行し、その結果を、前記公開鍵に対応する秘密鍵を用いて復号することによって、パラメータ a' を生成するステップと、

$r = h(a')$ を生成するステップと、
 r を前記クライアントに送信するステップとをさらに有することを特徴とする請求項 5 4 に記載の方法。

【請求項 5 7】 前記クライアントが、
a) $F_K(p)$ が前記公開鍵のメッセージ空間の要素であるかどうか、および、
b) $r = h(a)$ であるかどうかを決定するステップと、
a) または b) が真でない場合、前記クライアントが、認証を拒否するステップと、
a) および b) が真である場合、前記クライアントが、 $t = h'(a)$ を生成するステップと、
 t を前記サーバに送信するステップとをさらに有することを特徴とする請求項 5 6 に記載の方法。

【請求項 5 8】 前記サーバが、 $t = h'(a')$ であるかどうかを決定するステップと、
 $t = h'(a')$ である場合、前記サーバが、認証を容許するステップと、
 $t \neq h'(a')$ である場合、前記サーバが、認証を拒否するステップとをさらに有することを特徴とする請求項 5 7 に記載の方法。

【請求項 5 9】 前記サーバおよび前記クライアントが認証を受容した場合、前記サーバおよび前記クライアントが、後の安全な通信のためのセッション鍵を計算するステップをさらに有することを特徴とする請求項 5 8 に記載の方法。

【請求項 6 0】 前記クライアントが、前記公開鍵が前記テスト可能スーパーセットの要素であると決定した場合、
前記クライアントが、少なくとも前記公開鍵とパスワードの関数との関数としてパラメータ p を生成するステップと、

前記クライアントが、公開鍵空間マッピング関数 F_K を p に作用させた結果 $F_K(p)$ が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップとをさらに有することを特徴とする請求項 5 2 に記載の方法。

【請求項 6 1】 前記クライアントが、 $F_K(p)$ が前記公開鍵のメッセージ空間の要素であると決定した場合、

前記クライアントが、前記公開鍵を用いて前記公開鍵のメッセージ空間の実質的にランダムな要素 a を暗号化し、その結果と $F_K(p)$ との間で、公開鍵メッセージ空間の群演算を実行することによって、パラメータ q を生成するステップと、

q を前記サーバに送信するステップとをさらに有することを特徴とする請求項 6 0 に記載の方法。

【請求項 6 2】 前記クライアントが、 $F_K(p)$ が前記公開鍵のメッセージ空間の要素でない決定した場合、
前記クライアントが、パラメータ q を、前記公開鍵のメ

ッセージ空間の実質的にランダムな要素とするステップをさらに有することを特徴とする請求項 6 1 に記載の方法。

【請求項 6 3】 前記サーバが、 q が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、

前記サーバが、 q が前記公開鍵のメッセージ空間の要素でない決定した場合、前記サーバが、認証を拒否するステップと、

前記サーバが、 q が前記公開鍵のメッセージ空間の要素であると決定した場合、前記サーバが、前記公開鍵と前記パスワードの関数との関数としてパラメータ p' を生成するステップと、

前記サーバが、公開鍵空間マッピング関数 F_K を p' に作用させた結果 $F_K(p')$ が前記公開鍵のメッセージ空間の要素であるかどうかを決定するステップと、
前記サーバが、 $F_K(p')$ が前記公開鍵のメッセージ空間の要素でない決定した場合、前記サーバが、認証を拒否するステップと、

前記サーバが、 $F_K(p')$ が前記公開鍵のメッセージ空間の要素であると決定した場合、前記サーバが、 q と $F_K(p')$ との間で、公開鍵メッセージ空間の群演算の逆を実行し、その結果を、前記公開鍵に対応する秘密鍵を用いて復号することによって、パラメータ a' を生成するステップと、

$r = h(a')$ を生成するステップと、
 r を前記クライアントに送信するステップとをさらに有することを特徴とする請求項 6 1 に記載の方法。

【請求項 6 4】 前記クライアントが、
a) $F_K(p)$ が前記公開鍵のメッセージ空間の要素であるかどうか、および、
b) $r = h(a)$ であるかどうかを決定するステップと、

a) または b) が真でない場合、前記クライアントが、認証を拒否するステップと、
a) および b) が真である場合、前記クライアントが、 $t = h'(a)$ を生成するステップと、
 t を前記サーバに送信するステップとをさらに有することを特徴とする請求項 6 3 に記載の方法。

【請求項 6 5】 前記サーバが、 $t = h'(a')$ であるかどうかを決定するステップと、
 $t = h'(a')$ である場合、前記サーバが、認証を容許するステップと、

$t \neq h'(a')$ である場合、前記サーバが、認証を拒否するステップとをさらに有することを特徴とする請求項 6 4 に記載の方法。

【請求項 6 6】 前記サーバおよび前記クライアントが認証を受容した場合、前記サーバおよび前記クライアントが、後の安全な通信のためのセッション鍵を計算するステップをさらに有することを特徴とする請求項 6 5

に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ネットワーク認証および鍵交換に関し、特に、パスワード単独で (password-only) 安全な相互ネットワーク認証および鍵交換プロトコルに関する。

【0002】

【従来の技術】 ネットワークを通じての認証は、リモートクライアントがネットワークサーバにアクセスすることを可能にするシステムの安全性 (セキュリティ) の重要部分である。認証は一般に、次のうちの1つ以上を確認することによって実行される。

- ・ユーザが知っているもの (例えばパスワード)
- ・ユーザ自体、すなわち、指紋のような生物測定的情報
- ・ユーザが持っているもの、すなわち、スマートカードのような何らかの識別トークン

例えば、現金自動預入支払機 (ATM) は、これらのうちの2つ、すなわち、ユーザが持っているもの (ATMカード) と、ユーザが知っているもの (個人識別番号 (PIN: personal identification number) とを確認する。ATM認証は、データネットワークを通じての認証よりもずっと簡単である。その理由は、ATM自体が信頼されたハードウェアであるとみなされており、ATMカードの存在を確認し、正しい情報を安全に中央取引サーバに転送することが信頼されているからである。

【0003】 認証に加えて、鍵交換は、データネットワークを通じての通信の重要部分である。クライアントとサーバが認証された後、安全な通信チャネルをそれらの間に設定しなければならない。これは一般に、クライアントとサーバが、認証の後の通信中に使用するための鍵を交換することによって実行される。

【0004】 データネットワーク、特に、インターネットのような公衆データネットワークを通じての認証は困難である。その理由は、クライアントとサーバの間の通信が多くの異なるタイプの攻撃を受けやすいからである。例えば、盗聴攻撃では、敵は、クライアントとサーバの間の通信を傍受することによって秘密の情報を知ることがある。敵がパスワード情報を知った場合、敵は、その情報を再生 (リプレイ) して (リプレイ) してサーバに送って真正なクライアントになりすますことが可能となる (リプレイ攻撃 (replay attack) という)。リプレイ攻撃は、クライアントから送られたパスワードが暗号化されている場合でも有効である。その理由は、敵は、実際のパスワードを知る必要がなく、代わりに、サーバが真正なクライアントから期待するもの (この場合は、暗号化されたパスワード) をサーバに提供すればよいからである。もう1つのタイプの攻撃にスプーフィング (spoofing) 攻撃がある。この攻撃では、敵がサーバになりすまし、クライ

アントは、真正なサーバと通信していると信じるが、実際は、そうではなく敵と通信しているというものである。このような攻撃では、クライアントは敵に秘密情報を提供してしまうことがある。

【0005】 さらに、パスワードに基づく認証プロトコルでは、パスワードが弱く、辞書攻撃を受けやすい可能性がある。辞書攻撃とは、目的のパスワードに関するいくつかの既知情報に対して、多数の可能性があるパスワード (例えば、英語辞書内の全単語) をテスト (検査) することによって実行される、パスワードに対する力づくの攻撃である。既知情報は、例えば、公に入手可能な情報や、上記の方法のうちの1つより敵によって取得された情報である。ユーザはしばしば、容易に覚えられ、容易に推測されるパスワードを選択するため、辞書攻撃はしばしば有効である。

【0006】 ネットワーク認証にはさまざまな技術が知られている。これらの知られている技術は、2つの分類に分けることができる。第1の分類は、クライアントシステム上に永続的に記憶されたデータを必要とする技術からなる。第2の分類は、クライアントシステム上に永続的に記憶されたデータを必要としない技術からなる。

【0007】 第1の分類に関して、永続的に記憶されたデータには、決して暴露されてはならない秘密データ (例えば、認証サーバと共有する秘密鍵) や、秘密ではないが改竄されてはならない機密データ (例えば、認証サーバの公開鍵) がある。いずれのタイプの永続的データでも、敵からの攻撃からデータを守るために、追加の安全性が必要である。さらに、パスワードおよび永続的記憶データの両方に基づく認証プロトコルを使用する場合、一方に対する危険により、他方が攻撃を受けやすくなることがある。例えば、秘密鍵が危険にさらされると、パスワードに対する辞書攻撃が可能となる。この第1分類のプロトコルでのもう1つの問題は、永続的記憶データが鍵の生成および配送を必要とする点である。これは面倒なことがあり、一般に、システムの自由度が低くなる。

【0008】 第2分類は、パスワード単独認証プロトコルと呼ばれる。クライアントに永続的記憶データが不要であるからである。クライアントは、真正なパスワードを提供することができるだけでなく、潜在的に弱いパスワードを用いて強力な安全性および認証を提供するということは矛盾しているように思われるかもしれない。しかし、安全であるように設計されたパスワード単独ユーザ認証および鍵交換プロトコルがいくつか存在する。これらのプロトコルについては、D. Jablon, "Strong Password-Only Authenticated Key Exchange", ACM Computer Communication Review, ACM SIGCOMM, 26(5):5-20, 1996、に記載されている。これらのパスワード単独プロトコルのうち注目し値するものには以下のものがある。

- ・暗号化された鍵交換 (EKE: Encrypted Key Exchange)

ge) (S. M. Bellare and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks", Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 72-84, 1992, を参照)

・拡張EKE (A-EKE: Augmented-EKE) (S. M. Bellare and M. Merritt, "Augmented Encrypted Key Exchange: A Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise", Proceedings of the First Annual Conference on Computer and Communications Security, 1993, pages 244-250, を参照)

・修正EKE (M-EKE: Modified EKE) (M. Steiner, G. Tsudik, and M. Waidner, "Refinement and Extension of Encrypted Key Exchange", ACM Operating System Review, 29:22-30, 1995, を参照)

・単純パスワードEKE (SPEKE: Simple Password EKE) およびディフィ・ヘルマンEKE (DH-EKE: Diffie-Hellman EKE) (D. Jablon, "Strong Password-Only Authenticated Key Exchange", ACM Computer Communication Review, ACM SIGCOMM, 28(5):5-20, 1996, を参照)

・安全なリモートパスワードプロトコル (SRP: Secure Remote Password Protocol) (I. Wu, "The Secure Remote Password Protocol", Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium, pages 97-111, 1998, を参照)

・オープン鍵交換 (OKE: Open Key Exchange) (Stefan Lucks, "Open KeyExchange: How to Defeat Dictionary Attacks Without Encrypting Public Keys", Security Protocol Workshop, Ecole Normale Supérieure, April 7-9, 1997, を参照)

【0000】

【発明が解決しようとする課題】これらの知られているパスワード単独認証プロトコルの問題点は、これらが安全であることが証明されていないことである。実際、EKEプロトコルは、いくつかの数論的攻撃を受けやすい可能性がある (S. Patel, "Number Theoretic Attacks on Secure Password Schemes", Proceedings of the IEEE Symposium on Research in Security and Privacy", pages 236-247, 1997, を参照)。ネットワーク安全性の重要性に鑑み、安全であることが証明可能なパスワード単独相互認証プロトコルが必要とされている。

【0010】

【課題を解決するための手段】本発明は、公開鍵暗号方式を利用して安全なパスワード単独相互ネットワーク認証プロトコルを提供する。このプロトコルを実現するために用いられる特定の公開鍵暗号方式は、いわゆる使用可能 (usable) 暗号方式 (定義は後述) でなければならない。ネットワークサーバは、公開鍵暗号方式に従って公

開鍵・秘密鍵の対を生成し、公開鍵をクライアントに送信する。クライアントは、受信した公開鍵が、公開鍵暗号方式のすべての公開鍵の集合のいわゆるテスト可能スーパーセット (定義は後述) の要素であるかどうかを決定する。この決定が可能であるのは、公開鍵暗号方式が使用可能であることが要求されているからである。公開鍵がテスト可能スーパーセットの要素であるかどうかに関するクライアントによる決定は、クライアントに、サーバが適当な方法で選択した公開鍵を提供したかどうかを決定する方法を提供する。公開鍵がテスト可能スーパーセット内にないことがわかった場合、認証はクライアントによって拒否される。そうでない場合、プロトコルは続行される。

【0011】本発明の一実施例では、クライアントとサーバはいずれも、認証目的で使用される1つのパスワードを所有する。この実施例では、クライアントは、少なくとも公開鍵およびパスワードの関数としてパラメータ p を生成することによりプロトコルを続行する。公開鍵空間マッピング関数 F_K を p に作用させた結果 $F_K(p)$ が、公開鍵のいわゆるメッセージ空間の要素である場合、クライアントが、公開鍵を用いて公開鍵のメッセージ空間の現実的にランダムな要素を暗号化し、その結果と、 $F_K(p)$ との間で、公開鍵メッセージ空間の群演算を実行することによって、プロトコルは続行される。一方、 $e_K(p)$ がメッセージ空間の要素でない場合、クライアントは認証を拒否することを決定する。しかし、仮にクライアントがこの時点で拒否をサーバに通知したとすると、サーバはパスワードに関する有用な情報を抽出することができる可能性がある。そのため、クライアントは、認証を拒否することを選択しても、サーバに情報を漏らさないようにプロトコルを続行する。クライアントは、プロトコルの後のほうで、サーバがパスワードに関する有用な情報を得ることができなくなったときに、認証を拒否する。

【0012】本発明の第2実施例では、サーバにおける安全性の危険に対する保護のために、サーバは、パスワードを所有せず、その代わりにサーバには、パスワードの関数である値が提供され、サーバはそれを記憶する。パスワード自体は、サーバに記憶された値から決定することができる。

【0013】本発明の第3および第4の実施例は、使用可能公開鍵暗号方式としてRSA暗号方式を利用する。これらの実施例によれば、サーバが提供した公開鍵がすべてのRSA公開鍵の集合のテスト可能スーパーセットの要素であるかどうかを決定するためにRSA固有テストを行う。さらに、ある値が、RSAメッセージ空間の要素であるかどうかを決定するためにRSA固有テストを行う。第3実施例では、サーバが共有パスワードを記憶する。第4実施例では、サーバは、パスワードの関数である値を記憶する。

【0014】発明者は、本発明による相互認証プロトコルは、基礎となる公開鍵暗号方式と同程度に安全であることを証明した。従って、RSA固有の実施例では、発明者は、本発明のプロトコルがRSA暗号方式と同程度に安全であることを証明したことになる。その証明の概略は後述する。

【0015】

【発明の実施の形態】暗号は、二者間に安全な通信を提供する周知の技術である。本発明のさまざまな実施例について説明する前に、いくつかの背景知識および基礎的な用語について説明する。

【0016】まず、暗号方式について説明する。秘密鍵暗号方式では、メッセージ m は、暗号文 C を生成するために、暗号化関数 E および秘密鍵 K を用いて暗号化される。これは、 $C = E_K(m)$ と表される。暗号文 C は、秘密鍵 K を共有する二者間で安全に送信される。暗号文は、もとのメッセージ m を回復するために、復号関数 D および秘密鍵 K を用いて復号される。これは $m = D_K(C)$ と表される。

【0017】公開鍵暗号方式では、公開鍵(PK)および秘密鍵(SK)の対(PK, SK)が存在する。公開鍵は秘密ではなく、誰でも公開鍵を用いてメッセージ m を暗号化して $C = E_{PK}(m)$ となるような暗号文 C を生成することが可能である。暗号文は、秘密鍵を用いてのみ、 $m = D_{SK}(C)$ と復号可能である。暗号文は、公開鍵を用いては復号できない。公開鍵暗号は当業者に周知である。

【0018】周知の公開鍵暗号方式の1つにRSAがある。これは、R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signature and Public Key Cryptosystems", Communications of the ACM, vol. 21, 120-126, 1978,に記載されている。RSAでは、公開鍵は (N, e) であり、秘密鍵は (N, d) である。ただし、 N は、2つの大きいランダムに選択された素数 p と q の積であり(すなわち、 $N = p \cdot q$)、 e は、 e と $(p-1) \cdot (q-1)$ の最大公約数が1であるような2より大きい任意の数であり、 d は、 $e^{-1} \bmod (p-1) \cdot (q-1)$ である。暗号化関数は、 $E(m) = m^e \bmod N$ であり、復号関数は、 $D(C) = C^d \bmod N$ である。

【0019】次に、他のいくつかの暗号用語について説明する。非公式的には、集合 S から集合 T への関数 f が一方方向性関数であるとは、 S 内のすべての x に対して $f(x)$ を計算するのは容易であるが、 T 内のほとんどの y に対しては、 $f(x) = y$ であるような S 内の x を見つけることが計算量的に実現が可能であることである。一方方向性関数の一例は、法指数演算である。 p を大きい素数とし、 g を、 $\bmod p$ の乗法群(すなわち、 $1, \dots, p-1$ の範囲の数)の生成元とする。すると、 $f(x) = g^x \bmod p$ は一般に一方方向性関数

であると仮定される。その逆関数(離散対数関数という)は計算が困難である。また、離散対数関数の計算が困難であるその他の群(例えば、いくつかの楕円曲線群)もある。ディフィ・ヘルマン置換と呼ばれる置換プロトコル(W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, vol. 22, no. 6, 644-654, 1976,を参照)はこの関数に基づいている。具体的には、アリスとボブの二者が次のように秘密鍵を共有する。アリスが、ランダムな x を選択し、 $X = g^x \bmod p$ をボブに送る。一方、ボブは、ランダムな y を選択し、 $Y = g^y \bmod p$ をアリスに送る。秘密鍵を、アリスは $X^y \bmod p$ として、ボブは $Y^x \bmod p$ として、計算することができる($Y^x = X^y = g^{xy} \bmod p$ に注意)。また、ディフィ・ヘルマン置換は、離散対数関数の計算が困難であるその他の群(例えば、いくつかの楕円曲線群)上でも実行可能である。非公式的には、集合 S から集合 T への関数 h がランダムハッシュ関数であるとは、この関数が S 内の入力 x で計算されるまでは、 h の出力がランダムに見える。または、少なくとも予測不能であることをいう。一般にこのようにふるまう既知の関数は、SHA-1 (FIPS 180-1, "Secure Hash Standard", Federal Information Processing Standards Publication 180-1, 1995,を参照)、およびRIPEMD-160 (H. Dobbertin, A. Bosselaers, B. Preneel, "RIPEMD-160: a strengthened version of RIPEMD", in Fast Software Encryption, 3rd Intl. Workshop, pp. 71-82, 1996,を参照)である。

【0020】一般に、暗号方式は、その安全性のレベルを記述する安全性パラメータを有する。本明細書では、ハッシュ関数の安全性パラメータとして k を用い(ただし、 $1/2^k$ は無視できるほど小さいと仮定する)、公開鍵暗号方式の安全性パラメータとして l を用い、特に、RSAの法 N は長さ l ビットであると仮定する。

【0021】本発明の第1実施例による相互認証プロトコルを図1~図2に示す。図の左側に示すステップはサーバによって実行され、図の右側に示すステップはクライアントによって実行される。矢印は、クライアントとサーバの間の通信を表す。このプロトコルによれば、サーバは自分自身をクライアントに対して認証し、クライアントは自分自身をサーバに対して認証する。両側が認証をした後、それぞれ秘密鍵(セッション鍵という)を生成する。この鍵は、その後の安全な通信に使用可能である。

【0022】プロトコルを開始する前に、クライアントとサーバはある情報を所有していると仮定する。サーバは、使用される特定の公開鍵暗号方式に従って公開鍵・秘密鍵の対(PK, SK)を生成する。公開鍵・秘密鍵の対の生成は当業者に周知であり、ここでは説明しない。サーバとクライアントはいずれも、クライアントが

サーバとの認証に使用するパスワード π (すなわち、共有の秘密) を所有する。パスワード π は、クライアントとサーバの間で事前に設定されなければならない。各クライアント・サーバ対ごとに独立に、あるいは、各クライアント・サーバ対ごとに一意的であるように、選択されるべきである。

【0023】以下のプロトコルは、サーバおよびクライアントの両方を認証する。従って、サーバおよびクライアントはいずれも真正であるとは仮定されず、サーバまたはクライアントのいずれかが敵である可能性もある。クライアントは、自分自身を認証し、サーバにアクセスしようとしている敵である可能性がある。サーバは、疑いを持っていないクライアントから機密情報を得ようとして別の真正なサーバのふりをしよう (スプーフィング) としている敵である可能性がある。

【0024】当業者には直ちに明らかなように、サーバおよびクライアントは、コンピュータプログラムコードの制御下で動作するプログラムされたコンピュータとして実現可能である。コンピュータプログラムコードは、コンピュータ可読媒体 (例えばメモリ) に記憶され、コードは、コンピュータのプロセッサにより実行される。本明細書および図面が与えられれば、当業者は、ここで説明するプロトコルを実装するために適当なコンピュータプログラムコードを直ちに作成することが可能である。クライアントとサーバはデータネットワークを通じて互いに通信する。このようなネットワーク接続された、プログラムされたコンピュータは当業者に周知であり、ここではこれに詳細に説明しない。

【0025】図1～図2において、プロトコルが開始されると、ステップ110で、サーバは m を生成する。これは、集合 Ω のランダムな要素である。 Ω は、2つの同値な m の値を生成する確率が無視できることを保証するほど十分に大きな集合を表す。 Ω は、後の鍵交換を可能にするようなものとする。ステップ112で、サーバは、 m と PK をクライアントに送信する。上記のように、サーバは、プロトコルの開始前に (PK , SK) の対を生成していると仮定する。ステップ114で、クライアントは、ステップ112でサーバから受信した m が集合 Ω に属しているかどうか、および、 PK が集合 ε' (詳細は後述) に属しているかどうかを決定する。これらのテストのいずれかが偽である場合、クライアントは認証を拒否する。ステップ114のテストはクライアントによって実行される。その理由は、真正なサーバのふりをしている敵は、クライアントがプロトコルを実行した場合にこの敵がパスワード π に関する何らかの情報を知ることができるように m および PK を選択している可能性があるからである。

【0026】ここで、 PK が集合 ε' に属することの意味について説明する。上記のように、プロトコルが正しく動作するとともに機密情報を漏らさないために、公開

鍵・秘密鍵の対 (PK , SK) は、使用される特定の公開鍵暗号方式に従って適当に選択されなければならない。与えられた特定の公開鍵暗号方式に対して、ステップ112でサーバから受信した PK が、この特定の公開鍵暗号方式を用いて生成されるすべての可能な公開鍵の集合 ε の要素であるかどうかを決定することができれば理想的である。しかし、適当な時間内にこの決定をすることが可能な公開鍵暗号方式 (プロトコルに要求されるすべての性質を有する) は知られていない。そこで、使用可能公開鍵暗号方式およびテスト可能スーパーセット ε' を次のように定義する。

【0027】公開鍵暗号方式が使用可能であるとは、 ε の次のようなテスト可能スーパーセット ε' が存在することである。

1. すべての $PK \in \varepsilon'$ に対して、 $s_{PK} = |S_{PK}|$ は k に関して超多項式的である (すなわち、 S_{PK} の要素の個数は k の多項式ではおさえられない)。ただし、 S_{PK} は、 PK と、すべての可能なメッセージの暗号化とを用いて暗号化可能なすべての可能なメッセージの集合を表す。この集合 S_{PK} を、公開鍵 (PK) のメッセージ空間という。
2. 任意の PK に対して、 $PK \in \varepsilon'$ であるかどうかを決定する多項式時間アルゴリズムが存在する。
3. すべての $PK \in \varepsilon'$ に対して、 S_{PK} が一様に1つの要素を取り出す期待多項式時間アルゴリズムが存在する。
4. すべての $PK \in \varepsilon'$ に対して、任意の値 a について、 $a \in S_{PK}$ であるかどうかを決定する多項式時間アルゴリズムが存在する。
5. すべての $PK \in \varepsilon'$ に対して、整数 $\eta \geq 1 + k$ 、 $[0, 1]^\eta$ を定義域とする多項式時間計算可能な公開鍵空間マッピング関数 F_{PK} 、およびこの定義域の分割【数1】

$$X_{PK,1} \cup \dots \cup X_{PK,s_{PK}} \cup Z_{PK} \cup Z'_{PK}$$

(この分割は PK にのみ依存する) が存在して、以下の条件を満たす。

- (a) s_{PK} は多項式時間で計算可能である。
- (b) $i \in \{1, \dots, s_{PK}\}$ に対して、 $F_{PK}: X_{PK,i} \rightarrow S_{PK}$ は全単射である (すなわち、 F_{PK} は、各集合 $X_{PK,i}$ から S_{PK} への全単射を含む)。
- (c) それぞれの $a \in S_{PK}$ および $i \in \{1, \dots, s_{PK}\}$ に対して、 $F_{PK}(x) = a$ であるような $x \in X_{PK,i}$ を求める多項式時間アルゴリズムが存在する。
- (d) とえられた $x \in [0, 1]^\eta$ に対して、 $x \in Z_{PK}$ であるかそれとも $x \in Z'_{PK}$ であるかをテストする多項式時間アルゴリズムが存在する。
- (e) 各 $x \in Z_{PK}$ に対して、 $F_{PK}(x) \in S_{PK}$ でない。
- (f) $|Z'_{PK}|/2^\eta$ は、安全性パラメータ k に関して無視できる。

(g) $E \in \varepsilon$ である場合、 $\perp \in \mathbb{Z}_N \cup \mathbb{Z}'_N \mid \perp \neq 2$ は、安全性パラメータ k に関して無視される。

【0028】 当業者であれば、この使用可能な定義は、PK を用いて暗号化可能なメッセージの集合が、暗号化されたメッセージの集合に等しくないような暗号方式を含むように直ちに拡張することができる。当業者であれば、ここに記載したプロトコルを、このような暗号方式とともに使用するように直ちに修正することが可能である。こうして、公開鍵 PK に関して、ステップ 114 でのテストで、PK が、 ε のテスト可能スーパーセット ε' の要素であるかどうかを決定する。ステップ 114 でのテストが偽である（すなわち、 m は Ω の要素であり、かつ、PK は ε' の要素である）場合、認証は続行される。しかし、ステップ 114 でのテストが真である（すなわち、 m が Ω の要素でないか、または、PK が ε' の要素でない）場合、認証はクライアントによって拒否される。サーバは PK あるいは m を不適当な方法で選択したからである。

【0029】 ステップ 116 で、クライアントは、パラメータ μ を、集合 Ω のランダムな要素として設定する。ステップ 118 で、クライアントは、パラメータ a を、メッセージ空間 S_N のランダムな要素として設定する。ステップ 120 で、クライアントは、パラメータ (P, K, m, μ, π) のランダムハッシュ関数 H としてパラメータ p を計算する。ハッシュ関数 H は、十分なビット数（少なくとも n ）を出力する上記のような任意のランダムハッシュ関数とすることが可能である。

【0030】 ステップ 122 で、公開鍵空間マッピング関数 F_N を p に作用させた結果 $F_N(p)$ がメッセージ空間 S_N の要素であるかどうかを決定する。 $F_N(p)$ が S_N の要素でない場合、認証は拒否される。しかし、ステップ 122 で、 $F_N(p) \in S_N$ でない決定された場合、この時点で認証を終了するのは好ましくない。その理由は、クライアントがこの時点で認証を終了した場合、敵のサーバが、パスワード π についての何らかの知識を得る可能性があるからである。従って、クライアントは、認証を拒否すると決定した場合でも、サーバとのプロトコルを続行することが好ましい。そこで、ステップ 122 でのテストが真である場合、クライアントは $q = a$ と設定する。 $q = a$ と設定する（ a は、ステップ 118 で、メッセージ空間の実際のランダムな要素として選択された）ことによって、敵のサーバは、パスワード π に関する情報を得ることがなくなる。ステップ 122 で、 $F_N(p) \in S_N$ と決定された場合、クライアントは、 $q = E_N(a) \circ F_N(p)$ を計算することによって認証プロトコルを進める。すなわち、 a は公開鍵を用いて暗号化され、 $F_N(p)$ が、公開鍵メッセージ空間の群演算を用いて、暗号化結果に作用する。ステップ 124 で、 μ 、 q がサーバに送られる。

【0031】 ステップ 126 で、サーバは、 $\mu \in \Omega$ かつ

$q \in S_N$ であるかどうかを決定する。 $\mu \in \Omega$ でないか、または、 $q \in S_N$ でない場合、サーバは認証を拒否する。そうでない場合、ステップ 128 で、サーバは、パラメータ (P, K, m, μ, π) のランダムハッシュ関数 H としてパラメータ p' を計算する。このステップ 128 は、ステップ 120 について既に説明したのとに実行される。ステップ 130 で、サーバは、 $F_N(p')$ がメッセージ空間 S_N の要素であるかどうかを決定し、 S_N の要素でない場合、サーバは認証を拒否する。 $F_N(p')$ がメッセージ空間 S_N の要素である場合、認証は続行される。ステップ 132 で、サーバは、秘密鍵 S_K を用いて $q \circ F_N(p')$ を復号することによって a' を計算する（ここで \circ は、公開鍵メッセージ空間の群演算の逆を表す）。ステップ 134 で、サーバは、ランダムハッシュ関数 h を a' に作用させたものとして r を計算する。ステップ 136 で、サーバは r をクライアントに送る。

【0032】 ステップ 138 で、クライアントは、 $F_N(p) \in S_N$ かつ $r = h(a)$ であるかどうかを決定する。これらの条件が両方とも真である場合に限り、クライアントはサーバを真正であるとして受け入れる。なお、 $F_N(p) \in S_N$ でない場合には、クライアントは既にステップ 122 でサーバを受け入れないと決定しているが、敵がステップ 122 での認証の拒否から情報を得ることができないように認証プロトコルを続行していたことを想起すべきである。 $r = h(a)$ のテストは、サーバが正しいパスワード π を所有していたかどうかをテストする。そのため、ステップ 138 で、 $F_N(p) \in S_N$ でないか、または、 $r \neq h(a)$ であるためにクライアントが認証を拒否しても、サーバは、いかなる理由で認証が拒否されたかを決定することができない。ステップ 138 で、クライアントが、サーバを受け入れると決定した場合、ステップ 140 で、クライアントは、 $t = h'(a)$ を計算する。ただし、 h' はランダムハッシュ関数である。ステップ 142 で、クライアントは t をサーバに送る。

【0033】 ステップ 146 で、サーバは、 $t = h'(a')$ であるかどうかを決定する。 $t = h'(a')$ である場合、サーバは認証を受け入れる。そうでない場合、サーバは認証を拒否する。クライアントおよびサーバがいずれも認証を受け入れた場合、ステップ 144 で、クライアントはセッション鍵を計算し、ステップ 148 で、サーバはセッション鍵を計算する。セッション鍵は、共有秘密鍵として作用し、クライアントとサーバの間のその後の安全な通信のために使用される。このような方法での秘密鍵の使用は、公開鍵暗号の継続使用よりも、その後の安全な通信にとって効率的である。一実施例では、セッション鍵 K は、サーバおよびクライアントの両方で、 a のランダムハッシュ関数 h として、 $K = h(a)$ と計算される。代替実施例では、セッション

ン鍵Kは、サーバおよびクライアントの両方で、ディフィー・ヘルマンプロトコルを用いて、 m および μ をディフィー・ヘルマンパラメータに選択して、計算される。当業者には明らかなように、セッション鍵を計算するために、さまざまな代替法を使用可能である。

【0034】このようにして、図1～図2について説明したプロトコルは、上記で定義したように使用可能な公開鍵暗号方式を用いて、クライアントとサーバの相互認証を実現する。図1～図2について説明したプロトコルは、サーバがパスワード π を所有し記憶していることを仮定している。このようなプロトコルの1つの潜在的な問題点は、サーバ記憶領域の安全性が危険にさらされることにより、敵が、クライアントのパスワードを取得することが可能になることである。このような事態に対する保護のために、本発明の第2実施例では、サーバはパスワード π を所有せず、代わりに、パスワード π と $salt$ 値の関数である値 X を記憶する。 $salt$ 値は、敵が各 $salt$ 値ごとに別の辞書攻撃を実行するよう強制することによって、複数のパスワードに対する同時の辞書攻撃を妨げるために使用される公知の値である。値 X は、クライアントによってサーバに供給され、従って、サーバは、 X を知るのみであり、 X の知識から π を決定することはできない。クライアントは、 $X = g^x$ [X]として計算する。ただし、 g は離散対数の計算が困難な素数の生成元であり、 $x = H'(\pi, salt)$ であり、 H' は一方方向性ランダムハッシュ関数を表す。図1および図2に示した第1実施例について上記で説明したのと同様に、サーバは、使用される特定の公開鍵暗号方式に従って公開鍵・秘密鍵の対(PK, SK)を生成する。

【0035】以下、本発明の第2実施例によるプロトコルについて、図3～図4とともに説明する。ステップ205で、クライアントは、クライアントのユーザ名をサーバに送ることによってプロトコルを開始する。ステップ210で、サーバは、 Ω のランダムな要素として m を生成する。ステップ212で、サーバは、ステップ205で受信したユーザ名に対応する X および $salt$ を記憶領域から取得する。ステップ214で、サーバは、 m 、PKおよび $salt$ をクライアントに送る。ステップ216で、クライアントは、ステップ214でサーバから受信した m が集合 Ω に属しているかどうか、および、PKがテスト可能スーパーセット ε' に属しているかどうかを決定する。これらのテストのいずれかが偽である場合、クライアントは認証を拒否する。そうでない場合、認証は続行され、ステップ218で、クライアントは、ステップ214でサーバから受信した $salt$ を用いて $x = H'(\pi, salt)$ を計算する。ただし、 H' はランダムハッシュ関数である。ステップ220で、クライアントは、パラメータ μ を、集合 Ω のランダムな要素として設定し、ステップ222で、クライアント

は、パラメータ a を、メッセージ空間 S_m のランダムな要素として設定する。ステップ224で、クライアントは、パラメータ(PK, m , μ , g')のランダムハッシュ関数 H としてパラメータ p を計算する。このステップ224は、第1実施例のステップ120と同様であるが、この第2実施例では、 p を決定する際のパラメータの1つとしてパスワード π を使用する代わりに、パスワード π の関数、すなわち g' を、パラメータの1つとして使用する。その後、ステップ226、228、および230は、上記のステップ122、124、および126で説明したのと同様に実行される。

【0036】ステップ232で、サーバは、パラメータ(PK, m , μ , X)のランダムハッシュ関数 H としてパラメータ p' を計算する。このステップは、第1実施例のステップ128と同様であるが、ステップ232では、サーバは π を知らないので、代わりに、ハッシュ関数のパラメータとして X を使用する。ステップ234、236、および238は、上記のステップ130、132、および134で説明したのと同様に実行される。ステップ240で、集合 W のランダムな要素として y を選択する。ただし、 W は、相異なる g^r の値からなる十分に大きな集合を生じる g の可能な指数の集合を表す。ステップ242で、 $y = g^r$ と置く。ステップ244で、サーバは、 r および y をクライアントに送る。ステップ246は、上記のステップ138で説明したのと同様に実行される。ステップ248で、クライアントは $t = h'(a, y)$ を計算し、ステップ250で、クライアントは t をサーバに送る。

【0037】ステップ254で、サーバは、 $t = h'(a', X')$ であるかどうかを決定する。 $t = h'(a', X')$ である場合、サーバは認証を受け入れる。そうでない場合、サーバは認証を拒否する。クライアントおよびサーバがいずれも認証を受け入れた場合、ステップ252で、クライアントはセッション鍵を計算し、ステップ256で、サーバがセッション鍵を計算する。

【0038】図1～図2および図3～図4を参照して上記でそれぞれ説明した本発明の第1および第2実施例は、上記のように使用可能であるという要件を満たす公開鍵暗号方式とともに用いられる認証プロトコルである。このような使用可能な公開鍵暗号方式の1つは、パラメータを以下で説明するいくつかの制約に従って選択したRSA公開鍵暗号方式である。以下、本発明の第3および第4の実施例について説明する。第3実施例は、RSAを、サーバにパスワード π が記憶された公開鍵暗号方式として利用する。第4実施例は、値 X がサーバに記憶されたRSA公開鍵暗号方式を利用する。ただし、 X は、パスワード π と $salt$ 値の関数である。このように、第3および第4の実施例は、第1および第2実施例にそれぞれ対応するRSA固有の実施例である。

【0039】以下、本発明の第3実施例について、図5～図6とともに説明する。RSA公開鍵暗号方式では、公開鍵PKは2個のパラメータ(N, e)から構成され、秘密鍵SKは2個のパラメータ(N, d)から構成される。使用可能な形のRSA公開鍵暗号方式では、Nは大きく、eは、(N, e)を知っている者がNの任意の素因数rに対してeと(r-1)の最大公約数が1であることを容易にテストすることができるという性質を有することが保証されるように、公開鍵PK(N, e)は選択される。これを実行するいくつかの適当な方法について以下で説明する。サーバは、プロトコルの開始前に、適当な(PK, SK)の対を生成していると仮定する。ステップ302で、サーバは、Ωのランダムな要素としてmを生成する。ステップ304で、サーバは、m、N、およびeをクライアントに送信する。図1のステップ114について説明したように、クライアントは、次に、サーバから受信したmおよび公開鍵PKが、敵がパスワードπに関する情報を知り得るような方法でこれらの値を選択することに對する保護に適当であるように選択される。次に、ステップ306で、クライアントは、ステップ304でサーバから受信したmが集合Ωに属しているかどうか、および、PKがテスト可能スーパーセットε'（上記で定義）に属しているかどうかを決定する。RSAによる実装では、PKがテスト可能スーパーセットε'に属しているかどうかを決定する1つの方法は、Nおよびeが次の条件を満たすかどうかを決定することである。

$N \in [2^{1/2}, 2^1]$
 $e \in [2^1, 2^{1/2}]$
 eは素数である

Nおよびeが上記の条件をすべて満たす場合、PK ∈ ε'である。これらの条件は、Nが十分に大きい(Nが $2^{1/2} \sim 2^1$ の範囲内にあるかどうかを決定することにより)、および、eがNより大きい(eが $2^1 \sim 2^{1/2}$ の範囲内にあるかどうかを決定することにより)のテストを含む。図5のステップ306に示すいずれかの条件が偽である場合、認証は拒否される。そうでない場合、プロトコルはステップ308に進む。なお、RSA実施例では、PK ∈ ε'であるかどうかを決定するもう1つの代替テスト法がある。この代替テストは、次の条件が満たされるかどうかを決定することである。

$e \geq \sqrt{N}$
 $N \bmod e$ はNで割り切れない
 eは素数である

Nおよびeがこれらの条件をすべて満たす場合、PK ∈ ε'である。

【0040】なお、RSA固有の実施例では、PK ∈ ε'であるかどうかを決定するために使用可能な他のテスト法もある。例えば、eは素数であるかどうかをテストする代わりに、eを公知の固定値とし、eがこの固定

値に等しいことを確認するテストを行うことが可能である。当業者であれば、RSA固有の実施例でPK ∈ ε'であるかどうかを決定するための他のテストを実施することが可能である。

【0041】ステップ308、310、312はそれぞれ、上記の図1のステップ116、118、120で説明したのと同様に実行される。なお、ステップ120

(図1)における計算は、ステップ312(図5)に示したものと同じであるが、図1の一般的なPKは、図5～図6のRSA固有の実装ではNおよびeで置き換えられる。次のテストは、 $F_{PK}(p)$ が、RSA公開鍵のメッセージ空間の要素である(すなわち、 $F_{PK}(p) \in S_{PK}$)かどうかを決定する。RSA固有の実施例では、これは、pとNの最大公約数(gcd)が1に等しいかどうかを決定することによって実行される。gcd(p, N) = 1である場合、 $F_{PK}(p) \in S_{PK}$ であり、ステップ314で、クライアントは、 $q = (p \cdot a^e) \bmod N$ を計算する。gcd(p, N) ≠ 1である場合、 $F_{PK}(p) \in S_{PK}$ でないため、認証は拒否される。しかし、図1のステップ122に同じく既に述べたように、この時点で認証を終了するのは好ましくない。その理由は、そうすると、敵のサーバが、パスワードπについての何らかの知識を得る可能性があるからである。従って、クライアントは、認証を拒否すると決定した場合でも、サーバとのプロトコルを続行することが好ましい。そこで、ステップ314でのテストが真である場合、クライアントは、 $q = a$ とおくことにより、qをメッセージ空間の実質的にランダムな要素に設定する。ステップ316で、クライアントは、μ、qをサーバに送信する。

【0042】ステップ318で、サーバは、 $\mu \in \Omega$ かつ $q \in S_{PK}$ であるかどうかを決定する。 $q \in S_{PK}$ である(すなわち、qがRSA公開鍵のメッセージ空間の要素である)かどうかのテストは、gcd(q, N) = 1であるかどうかをテストすることにより行われる。 $\mu \in \Omega$ でないか、または、gcd(q, N) ≠ 1である場合、サーバは認証を拒否する。そうでない場合、ステップ320で、サーバは、パラメータ(N, e, m, μ, π)のランダムハッシュ関数Hとしてパラメータp'を計算する。ステップ322で、サーバは、gcd(p', N) = 1であるかどうかを決定することにより、 $F_{PK}(p')$ がRSA公開鍵のメッセージ空間の要素であるかどうかを決定する。gcd(p', N) ≠ 1である場合、 $F_{PK}(p')$ はRSA公開鍵のメッセージ空間の要素ではなく、サーバは認証を拒否する。gcd(p', N) = 1である場合、 $F_{PK}(p')$ はRSA公開鍵のメッセージ空間の要素であり、認証は続行される。ステップ324で、サーバは、(q/p')に対してRSA復号を実行する。ステップ326～340は、それぞれ図2のステップ134～148と同様に実行さ

れる。なお、この第3実施例はRSA固有であるが、 $F_{pk}(p) \in S_{pk}$ であるかどうかを決定するステップ330(図2のステップ138に対応する)におけるテストは、 $gcd(p, N) = 1$ であるかどうかを決定することにより実行される。

【0043】本発明の第4実施例について図7～図8を参照して説明する。サーバは、パスワード π および s_{alt} 値の関数である値 x を記憶する。これは第2実施例と同様である。しかし、第2実施例は任意の使用可能な公開鍵暗号方式を用いたプロトコルであった。第4実施例は、公開鍵暗号方式としてRSAを使用する。図7において、ステップ402で、クライアントは、ユーザ名をサーバに送ることによってプロトコルを開始する。ステップ404で、サーバは、 Ω のランダムな要素として m を生成する。ステップ406で、サーバは、ステップ402で受信したユーザ名に対応する X および s_{alt} を記憶領域から取得する。ステップ408で、サーバは、 m 、 N 、 e 、および s_{alt} をクライアントに送る。ステップ410で、クライアントは、ステップ408でサーバから受信した m が集合 Ω に属しているかどうか、および、公開鍵 N 、 e がテスト可能スーパーセット e' に属しているかどうかを決定する。このRSA固有の実施例では、公開鍵がテスト可能スーパーセット e' に属しているかどうかのステップ410におけるテストは、上記の図5とともに説明したステップ306におけるテスト(または、上記のステップ306とともに説明した代替テスト)と同じである。ステップ412、414、および416で、クライアントは、それぞれ上記の図3のステップ218、220、および222とともに説明したようにして、パラメータ x 、 μ 、 a を生成する。ステップ418で、クライアントは、パラメータ (N, e, m, μ, g^x) のランダムハッシュ関数 H としてパラメータ p を計算する。次に、ステップ420で、クライアントは、 $gcd(p, N) = 1$ であるかどうかを決定することによって、 $F_{pk}(p) \in S_{pk}$ であるかどうかを決定し、上記の図5のステップ314とともに説明したのと同様に適当にパラメータ q を生成する。その後、ステップ422で、クライアントは、 μ 、 q をサーバに送信する。

【0044】ステップ424で、サーバは、 $\mu \in \Omega$ かつ $q \in S_{pk}$ であるかどうかを決定する。 $q \in S_{pk}$ であるかどうかのテストは、 $gcd(q, N) = 1$ であるかどうかをテストすることにより行われる。 $\mu \in \Omega$ でないか、または、 $gcd(q, N) \neq 1$ である場合、サーバは認証を拒否する。そうでない場合、ステップ426で、サーバは、パラメータ (N, e, m, μ, X) のランダムハッシュ関数 H としてパラメータ p' を計算する。その後、ステップ428、430、432は、それぞれ図5～図6のステップ322、324、326と同様に実行され、ステップ434、436は、それぞれ図4のステ

ップ240、242と同様に実行される。

【0045】ステップ438で、サーバは、 r および y をクライアントに送る。ステップ440～450は、それぞれ図4のステップ246～256と同様に実行される。ステップ440の $F_{pk}(p) \in S_{pk}$ であるかどうかのテストは、 $gcd(p, N) = 1$ であるかどうかをテストすることにより、RSA固有の方法で実行される。

【0046】本発明の発明者は、本発明による相互認証プロトコルは、基礎となる公開鍵暗号方式と同程度に安全であることを証明した。従って、RSA固有の実施例では、発明者は、本発明のプロトコルがRSA暗号方式と同程度に安全であることを証明したことになる。その証明の概略を述べる。

【0047】本発明が安全な相互認証プロトコルであることを証明するため、プロトコルの安全性の議論を、使用される暗号化関数の安全性の議論に帰着させる。具体的には、入力として暗号化関数および暗号文をとるシミュレータを考え、以下の事象のうちの1つが起こるように、敵に対してランダムに選択されたパスワードでプロトコルをシミュレートする。

1. いくつかのランダムに選択された値が衝突する(すなわち、等しくなる)ことがまれに起こる。これは、非常に低い確率 β で起こることが示される。
2. シミュレータは、無視できない確率で、暗号文に対する復号を導出できるという事象がある。
3. 上記以外。この場合、敵がプロトコルを破る確率は高々 ν/d であることを直接に証明する。ただし、 ν はアクティブな「スプーフィング」攻撃の数であり、 d は可能なパスワードの数である。

【0048】ここで、敵が、無視できない ε に対して、確率 $\nu/d + \varepsilon$ でプロトコルを破るとする。(非公式的には、これは、敵が、単にパスワードを推測してそれぞれでログインしようとするより実質的に高い確率でプロトコルを破ることができることを意味する。これは不可能であることをわれわれは主張する。)この敵を用いると、入力として暗号化関数および暗号文をとり、無視できない確率で暗号文を復号するアルゴリズム A を構成することができることになる。 A は、シミュレータに対する敵を実行する。 E_1 、 E_2 および E_3 を上記の3つの事象とし、 B を、敵がプロトコルを破る事象とする。上記の議論から、次のことがわかる。

【数2】

$$\begin{aligned}
\frac{v}{d} + \varepsilon &\leq \Pr(B) \\
&\leq \Pr(E_1) + \Pr(B \wedge E_1) \\
&= \Pr(E_1) + \Pr(B \wedge E_1 \wedge \overline{E_2}) + \Pr(B \wedge \overline{E_1} \wedge \overline{E_2}) \\
&\leq \Pr(E_1) + \Pr(E_2) + \Pr(B \wedge E_2) \\
&= \Pr(E_1) + \Pr(E_2) + \Pr(B|E_2) \cdot \Pr(E_2) \\
&\leq \Pr(E_1) + \Pr(E_2) + \Pr(B|E_2) \\
&\leq \beta + \Pr(E_2) + \left(\frac{1}{d}\right)
\end{aligned}$$

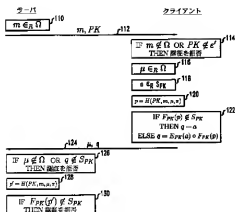
従って、 $\Pr(E_2) \geq \varepsilon - \beta$ であり、これは無視できない。従って、Aは、無視できない確率で暗号文を復号することができるが、これは、暗号化関数の安全性と矛盾する。

【0049】

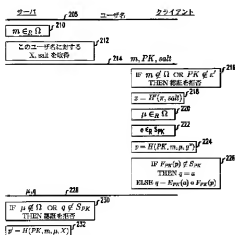
【発明の効果】以上述べたごとく、本発明によれば、安全であることが証明可能なパスワード単独相互認証プロトコルが得られる。

【図面の簡単な説明】

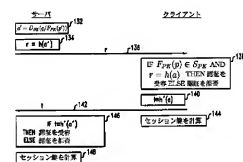
【図1】



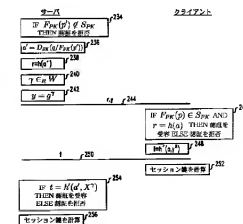
【図3】



【図2】



【図4】



【図1】サーバがパスワードを記憶する認証プロトコルの実施例を示す図である。

【図2】サーバがパスワードを記憶する認証プロトコルの実施例を示す図である。

【図3】サーバが、パスワードの関数である値を記憶する認証プロトコルの実施例を示す図である。

【図4】サーバが、パスワードの関数である値を記憶する認証プロトコルの実施例を示す図である。

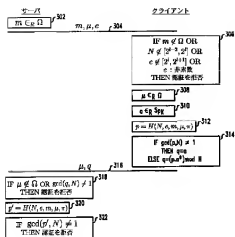
【図5】サーバがパスワードを記憶する認証プロトコルのRSA固有の実施例を示す図である。

【図6】サーバがパスワードを記憶する認証プロトコルのRSA固有の実施例を示す図である。

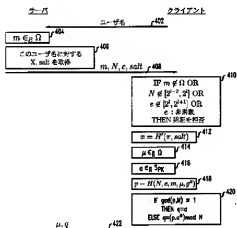
【図7】サーバが、パスワードの関数である値を記憶する認証プロトコルのRSA固有の実施例を示す図である。

【図8】サーバが、パスワードの関数である値を記憶する認証プロトコルのRSA固有の実施例を示す図である。

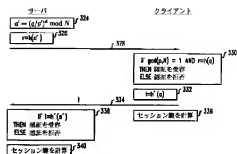
【図5】



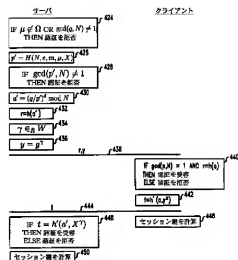
【図7】



【図6】



【図8】



フロントページの続き

(71) 出願人 596077259

600 Mountain Avenue,
Murray Hill, New Je
rsey 07974-0636 U. S. A.

(72) 発明者 フィリップ ダグラス マッケンジー
アメリカ合衆国、07040 ニュージャージー
一、メイブルウッド、カーレトン コート
11

(72) 発明者 ラム スワミナサン
アメリカ合衆国、07974 ニュージャージー
一、ニュー プロビデンス、ガレス ドラ
イブ #3 72